

C++模拟试卷-2

一、选择题（每小题3分，共15分）

1. 类 A 是一个包含纯虚函数的抽象类，下列说明语法正确的是 **C**

A. A a; B. A f();
C. A &f(); D. A f(A);

2. 使用 `exit` 退出程序，关于对象自动析构，哪个叙述正确 **B**

A.不析构全局对象但析构局部对象
B.析构全局对象但不析构局部对象
C.全局对象和局部对象都不析构
D.全局对象和局部对象都析构

3. 对于如下程序:

```
#include <stdio.h>

struct A {
    A() { printf("别理我"); }
    A(char *s) { printf(s); }
} a("烦着呢");

A f() { printf("一边去"); return a; }

void main(void) { A f(); }
```

A. 输出为“烦着呢”;
B. 输出为“烦着呢 别理我”;
C. 输出为“一边去”;
D. 输出为“烦着呢 一边去”;

4. 对于 `int x`; 如下运算错误的是 **A**

A. $x^{++ ++}$ B. $++ ++x$
C. $(++x)++$ D. $++ (++x)$

5. 对于 `int x`, `int &y` 最好引用如下哪个表达式 A

A. $x+=3$; B. $x+4$;
C. $x++$; D. $(-x)--$;

二、指出各类可访问的成员及其访问权限 (20)。

```
class A {
    int a;
protected:
    int b, f;
public:
    int c, d;
```

```
};
class B: protected A {
    int d;
protected:
    int c, e;
public:
    int f;
};
```

```
class C: public A {
    int g;
protected:
    int h, d;
public:
    int c, i;
};
```

```
struct D: B, public C {
    int j;
protected:
    int k, c;
public:
    int n;
};
```

类A: private: a

protected: b, f

public: c, d

类B: private: d

protected: c, e, A::b, A::f, A::c, A::d

public: f

类C: private: g

protected: h, d, A::b, A::f

public: c, i, A::c, A::d

类D: private:

protected: k, c, B::c, B::e, B::A::b, B::A::f, B::A::c, B::A::d, C::h, C::d, C::A::b, C::A::f

public: j, n, B::f, C::c, C::i, C::A::c, C::A::d

三、指出 main 中每行的输出结果（前四题每题 3 分，后2题每题4分）

```
#include <iostream>
```

```
struct A { A() { cout << 'A'; } };
```

```
struct B { B() { cout << 'B'; } };
```

```
struct C: A { C() { cout << 'C'; } };
```

```

struct D: B, virtual C { D() { cout << 'D'; } };
//struct E: virtual A, virtual D {
struct E: A, virtual D {
    D d;
    E(): A() { cout << 'E'; }
};
struct F: B, virtual C, E, virtual D {
    D d;
    F() { cout << 'F'; }
};

void main() {
    A a; cout << "\n";    //输出=A
    B b; cout << "\n";    //输出=B
    C c; cout << "\n";    //输出=AC
    D d; cout << "\n";    //输出=ACBD
    E e; cout << "\n";    //输出=ACBDAACBDE
    F f; cout << "\n";    //输出=ACBDBAACBDEACBDF
}

```

四、指出以下程序的语法错误及其原因（每错约1分，共15分）

```

class A {
    int a;
protected:
    const int &b;
    ~A() {} //错，A类的析构函数为protected，生成对象后不可访问析构函数，无法析构
public:
    int c;
    virtual A (*g)(int); //错，不能将函数指针声明为virtual
    A(int x) { a = x; }; //错，b是const int变量，应放在初始化列表中初始化
} x = (4, 3);

class B: A {
    int d;
public:
    A::b;
    friend int operator()(int) { return 2; }; //错，()只能重载为实例函数成员，不能声明为友元
    B(int x, int y, int z) { d = x + y + z; };
} b(5, 6, 7);

struct C: B {
    int z;
public:

```

```

    ~C(int x) { z = x; };
} c;

void main() {
    int A::*p = &c.z; //错, p是指向A类对象中int变量的指针, 而c不是A类对象
    int i = x.b; //错, x.b为protected, 不可访问
    i = x; //错, A类没有定义int强制类型转换重载函数
    i = b.b;
    i = i + c.d; //错, B::d为private, 不可访问, 无法被C类继承
    i = b.*p; //错, 成员指针不能参与运算
}

```

五、请填写入学号最后一位十进制数字，指出 main 函数中变量 i 在每条赋值语句执行后的值（每小题 2.5 分，共 15 分）

```

int x = __2__ //学号最后一位十进制数
int y = x + 3;
struct A {
    int x;
    static int &y;
public:
    operator int() const { return x + y; }
    int &v(int &x) {
        for(int y = 1; x < 301; x ^= y++) {
            if(x > 300) { x -= 31; y = 2; }
        }
        return ++x;
    }
    A &operator++() { ++x; ++y; return *this; }
    A(int x = A::y + 2, int y = ::x + A::y) { A::x = x + 1; A::y = y + 2; }
};

int &A::y = ::y;

void main() {
    A a(3, 4), b(a), c;
    int i, &j = i;
    int A::*p = &A::x;
    j = a.x; // i = 4
    i = a.y; // i = 12
    i = a.*p; // i = 4
    i = ++a; // i = 19
    i = b.y + ::y; // i = 18
}

```

```

        (b.v(i)=3) += 2;        // i = 5
    }

```

六、一个自然数，如果它等于除其本身之外的所有其它不同因子之和，则这个自然数被称为完美数。例如， $6=1+2+3$, $28=1+2+4+7+14$ 。试编写如下完美数类中的所有函数成员的函数体代码（每小题 2.5 分，共 15 分）

```

class PER {
    const int n;           //存放自然数
    int *const f;           //存放所有有效因子,所有因子之和等于 n 才是完美数
    int c;                 //有效因子个数：正数表示是完美数，负数表示不是
public:
    PER(int p);             //用自然数 p 初始化 n, f, c
    PER(const PER &p);      //深拷贝构造函数
    PER &operator=(const PER &p); //深拷贝赋值运算
    operator int() const;    //若不是完全数则返回 0，否则返回 c
    operator [ ] (int k) const; //返回 k 所指示的因子,若 k < 0 或 k >= c 返回 0
    ~PER();                 //析构函数
};

```

//提示: f 分配的整型内存单元数量不会超过 $n/2$ 。

```

PER::PER(int p) :n(p), f(new int[p / 2]) {
    c = 0;
    for (int i = 1; i <= p / 2; i++)
        if (!(n % i))
            f[c++] = i;
}

```

```

PER::PER(const PER& p) :n(p.n), f(new int[p.n / 2]) {
    c = p.c;
    for (int i = 0; i < c; i++)
        f[i] = p.f[i];
}

```

```

PER& PER::operator=(const PER& p) {
    *(int*)&n = p.n;
    *(int**)&f = new int[p.n / 2];
    c = p.c;
    for (int i = 0; i < c; i++)
        f[i] = p.f[i];
}

```

```
PER::operator int() const {  
    int count = 0;  
    for (int i = 0; i < c; i++)  
        count += f[i];  
    if (count == n)  
        return 1;  
    else  
        return 0;  
}
```

```
int PER::operator [ ] (int k) const {  
    if (k < 0 || k >= c)  
        return 0;  
    else  
        return f[k];  
}
```

```
PER::~~PER() {  
    *(int*)&n = 0;  
    c = 0;  
    if (f)  
        delete[]f;  
}
```